



# TREBALL FINAL DE GRAU



ESCOLA  
POLITÈCNICA SUPERIOR  
UNIVERSITAT DE LLEIDA  
INSPIRING THE FUTURE

Estudiant: Guillem Felis de Dios

Titulació: Grau en Enginyeria Informàtica

Títol de Treball Final de Grau: **Detecció de malalties en vegetals mitjançant xarxes neuronals**

Director/a: **Josep Argelich Romà**

Presentació

Mes: Juliol

Any: 2021

# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
1.1	Motivació del projecte . . . . .	2
1.2	Objectius . . . . .	2
1.3	Estructura del document . . . . .	2
<b>2</b>	<b>Marc teòric: Què és la IA?</b>	<b>4</b>
2.1	Origen . . . . .	4
2.2	Categories . . . . .	5
2.2.1	IA forta . . . . .	5
2.2.2	IA dèbil . . . . .	5
2.3	Conceptes clau . . . . .	5
2.3.1	IA . . . . .	6
2.3.2	Machine Learning . . . . .	6
2.3.3	Deep Learning . . . . .	6
2.4	Xarxes neuronals . . . . .	7
2.4.1	Neurones . . . . .	7
2.4.2	Xarxa neuronal profunda . . . . .	8
2.4.3	Xarxa neuronal convolucional . . . . .	9
2.5	Fases d'un projecte de <i>Machine Learning</i> . . . . .	10
2.6	Aplicacions en <i>Computer Vision</i> . . . . .	11
<b>3</b>	<b>Disseny del projecte</b>	<b>13</b>
3.1	Anàlisi de Requeriments . . . . .	13
3.2	Obtenció de dades . . . . .	13
3.3	Entorns d'entrenament . . . . .	14
3.3.1	Local . . . . .	14
3.3.1.1	Combinació de CPU i GPU . . . . .	14
3.3.1.2	Nvidia Jetson Nano 2GB TPU . . . . .	15
3.3.2	Cloud . . . . .	15
3.3.2.1	Google colab . . . . .	15
3.3.2.2	Google cloud . . . . .	16
3.3.2.3	Kaggle . . . . .	16
<b>4</b>	<b>Anàlisi exploratori de les dades</b>	<b>17</b>
4.1	Distribució de les classes . . . . .	17
4.2	Visualització de mostres . . . . .	19

<b>5</b>	<b>Preprocessament</b>	<b>20</b>
5.1	Normalització . . . . .	20
5.2	Eliminació de duplicats . . . . .	20
5.3	Ajustar la resolució de les imatges . . . . .	21
5.4	Preparació dels conjunts de test, entrenament i validació . . . . .	21
5.5	Data Augmentation . . . . .	21
<b>6</b>	<b>Modelat</b>	<b>22</b>
6.1	MaxPooling . . . . .	22
6.2	ReLU . . . . .	23
6.3	DenseNet . . . . .	23
6.3.1	Arquitectura del model . . . . .	23
6.3.2	Resultats . . . . .	25
6.4	EfficientNet . . . . .	29
<b>7</b>	<b>Conclusions i futur del projecte</b>	<b>31</b>
<b>8</b>	<b>Bibliografia</b>	<b>33</b>

# Índex de figures

2.1	IA vs ML vs DL . . . . .	6
2.2	Exemple d'una neurona . . . . .	7
2.3	Estructura d'una Xarxa Neuronal . . . . .	8
2.4	Filtre de convolució . . . . .	9
2.5	Pèrdua d'informació d'un kernel de mida 2x2, a l'esquerra la imatge original i a la dreta la imatge filtrada . . . . .	10
2.6	Fases bàsiques d'un projecte de ML . . . . .	10
4.1	Distribució inicial . . . . .	17
4.2	Precisió amb la distribució inicial . . . . .	18
4.3	Distribució corregida . . . . .	19
4.4	Mostres per classe . . . . .	19
6.1	Maxpooling amb una finestra de 2x2 . . . . .	22
6.2	Activació per ReLU . . . . .	23
6.3	Blocs bàsics DenseNet121 . . . . .	24
6.4	Arquitectura del model final . . . . .	25
6.5	Evolució en la precisió sobre els conjunts d'entrenament i validació . . . . .	25
6.6	Classificació inicial, llindar del 55% . . . . .	26
6.7	Distància relativa, llindar del 20% i 10% respectivament . . . . .	27
6.8	Mostra d'imatges duplicades . . . . .	27
6.9	Rendiment durant l'entrenament de DenseNet, sense duplicació . . . . .	28
6.10	Rendiment sobre el conjunt de test . . . . .	29
6.11	Evolució de la precisió sobre els conjunts d'entrenament i validació d'EfficientNetB7 . . . . .	30

# Índex de taules

3.1	Comparativa d'entorns d'entrenament . . . . .	16
7.1	Evolució dels resultats . . . . .	32

## Abstract

Every year crops around the world are being destroyed due to plagues and diseases, making the urge to find an effective way to fight against them more and more important. With the recent advancements in the field of Artificial Intelligence and, more concretely, in Computer Vision, this project aims to develop a model able to classify some illnesses that are visible on apple leaves. To do so, DensetNet and EfficientNet pre-trained models have been used, re-trained with the data from the Plant Pathology 2021 - FGCv8 open competition. Furthermore, this document also presents the results of performed experimentation that shows how variation on certain hyperparameters affects the performance of such models. Finally, the resulting trained model has a tested accuracy of 87.36%, performing extremely well on single-class classification, with a tested accuracy of 91.26% in that case.

## Resum

Cada any collites d'arreu del món són destruïdes a causa de plagues i malalties, cosa que fa que cada vegada sigui més important la voluntat de trobar una manera eficaç de lluitar contra elles. Amb els recents avenços en el camp de la Intel·ligència Artificial i, més concretament, en Visió Artificial, aquest projecte pretén desenvolupar un model capaç de classificar algunes malalties visibles en fulles de poma. Per fer-ho, s'han utilitzat els models pre-entrenats DensetNet i EfficientNet, entrenats de nou amb les dades del concurs obert Plant Pathology - FGCv8. A més, el document també presenta els resultats de l'experimentació realitzada, que mostra com la variació de certs hiperparàmetres afecta el rendiment d'aquests models. Finalment, el model entrenat resultant ha obtingut una precisió comprovada de 87.36%, amb un rendiment especialment bo per la classificació d'etiqueta única, on ha obtingut un 91.26% de precisió.

# 1. Introducció

L'agricultura té un paper crucial en l'economia global, sobretot en els països en vies de desenvolupament ja que proporciona la principal font d'aliments, ingressos i ocupació a les seves poblacions rurals. Segons la FAO (*Food and Agriculture Organization of the United Nations*), s'ha establert que el valor afegit mundial en l'agricultura ha crescut un 68% entre el 2000 i el 2018, amb un valor total de 3.4 trilions de dòlars, representant sobre un 4% del PIB mundial des de l'any 2000, tot i que en països en desenvolupament pot arribar a representar fins a més d'un 50% del PIB nacional. Per exemple, al Txad l'agricultura representa un 57.2% del PIB mentre que a Espanya no arriba al 3%.

Actualment, aquests models econòmics es troben en risc per diversos factors com el canvi climàtic, l'escassetat d'aigua i la propagació de plagues, així com factors polítics i marges de benefici molt ajustats. De fet, un estudi publicat l'any 2019 a la revista *Nature, Ecology & Evolution* [1], estima que les plagues i els patògens van causar unes pèrdues d'entre el 17% i 30% de mitja (depenent del conreu) en la producció dels principals productes de consum al món, com el blat de moro, l'arròs, el blat i la patata.

Paral·lelament, el sector tecnològic s'està apropant cada cop més a l'agrícola, on han guanyat popularitat les empreses anomenades *d'agricultura de precisió o intel·ligent* que estan aplicant tecnologies com la IA (*Intel·ligència Artificial*) o el IoT (*Internet of Things*) per tal d'intentar solucionar alguns d'aquests problemes.

Alguns exemples d'aquestes d'empreses emergents en el sector anomenat *AgTech* són Cropin, Arable o CropX, que ja ofereixen eines de gestió agrícola basades en IoT per tal d'adquirir dades amb les que aconseguixen realitzar els seguiments pertinents del conreu d'una forma precisa, i a partir de les quals també, gràcies a la IA, són capaços de realitzar prediccions sobre el futur del conreu o del clima, així com de possibles deficiències presents en el cultiu.

A més, durant l'any 2019, gegants de la tecnologia com Nvidia i Google van començar a comercialitzar components enfocats als desenvolupadors, les anomenades TPUs (*Tensor Process Unit*), que permeten augmentar altament la eficiència en el processament de dades, i per tant, possibiliten la creació de noves aplicacions de projectes basats en IA.

Aquest tipus de components fan possible projectes com el de l'empresa *FJDynamics* i el seu producte *Auto Steering Kit*, un projecte pioner que amb l'ajuda d'una càmera d'IA, un volant elèctric, sensors i una antena 4G és capaç de convertir qualsevol tractor comercial en un tractor amb mode de conducció automàtica, on el conductor només ha de reduir la velocitat per tal que el sistema giri el tractor per sí sol.

## 1.1 Motivació del projecte

Inicialment, el projecte és va dissenyar amb l'objectiu de combatre una patologia molt present en la major part dels països europeus i del mediterrani occidental, el foc bacterià (*Erwinia Amylovora*). Afecta a diversos vegetals de la família de les rosàcies, entre els quals es troben la perera (*Pyrus*), la pomera (*Malus*), el codonyer (*Cydonia*) i els nesprers (*Eryobotrya i Mespilus*) així com plantes ornamentals i silvestres.

És una malaltia d'una alta importància econòmica, ja que afecta a espècies d'un gran valor comercial, és altament contagiosa i devastadora, no existeixen mètodes de control efectius ni tractaments químics curatius i està considerada com un organisme nociu de quarantena a la Unió Europea. El tractament actual és l'eliminació completa dels espècimens afectats per tal d'evitar-ne la propagació. A Catalunya la seva afectació es troba concentrada principalment en el correu de pera en la província de Lleida.

Per aquests motius, la idea en què es basa el projecte és la creació d'un model d'IA capaç de detectar la presència de patologies en vegetals a partir d'imatges, per tal d'augmentar la velocitat de la diagnosi i mitigar-ne, per tant, els possibles efectes nocius.

## 1.2 Objectius

A continuació es descriuen els diferents objectius del projecte:

- Classificar i definir el problema a resoldre, per decidir quin tipus de dades i tècniques són necessàries utilitzar per donar-hi sol·lució.
- Trobar un conjunt de dades de qualitat que permeti la implementació d'un model.
- Implementar un model efectiu capaç de diagnosticar les malalties que finalment s'hagin escollit.
- Avaluar el model implementat finalment, comprovant que és efectiu per resoldre el problema i justificar-ne la seva validesa a partir de paràmetres mesurables.

## 1.3 Estructura del document

En el primer capítol s'introdueix la idea a partir de la qual neix el projecte, així com la motivació que l'ha originat. També es defineixen els diferents objectius a complir durant el desenvolupament del mateix.

En el següent capítol, s'explicaran els diferents conceptes teòrics essencials per tal d'entendre el context científic en el qual s'engloba el projecte i el desenvolupament del mateix, així com diversos termes que s'utilitzaran al llarg del document. Es parlarà sobre els orígens de la IA, com la podem definir, com s'estructura un projecte de *machine learning* i quines són les diverses aplicacions de la IA.

El tercer capítol tracta del disseny del projecte: es defineixen els requeriments observables que permetran donar resposta als objectius marcats en el primer capítol i s'expliquen les diferents decisions que s'han pres en termes de disseny per tal d'adequar-ne la viabilitat del desenvolupament. En aquest capítol es presenta el conjunt de dades que s'ha triat finalment així com els entorns d'entrenament que s'han valorat per desenvolupar el model.

A continuació, es comença pròpiament amb el procés de desenvolupament del model d'IA, analitzant les dades inicials amb l'objectiu d'entendre la composició i les característiques del conjunt d'entrada.

Seguidament, en el capítol de preprocessament es mostren quines tècniques han sigut necessàries aplicar per tal de preparar les dades abans de ser tractades.

En el següent capítol, el modelat, s'acaben d'explicar els últims conceptes teòrics necessaris per entendre l'arquitectura del model utilitzat finalment, així com els resultat que s'han observat durant la fase d'entrenament en diversos models. També es mostra l'evolució del rendiment del model final a partir de l'aplicació de tècniques que n'han millorat la precisió gradualment.

Finalment, es sintetitzen els resultats obtinguts i com hauria d'evolucionar el projecte en el futur en l'últim capítol de conclusions.



## 2. Marc teòric: Què és la IA?

Per tal de poder parlar de les necessitats concretes d'aquest projecte en el següent capítol, és important entendre què és la Intel·ligència Artificial, així com les característiques generals de la metodologia que segueixen els projectes pràctics d'aquest camp.

Cal dir, a més, que no existeix una definició acceptada de forma universal, tot i així, es pot donar resposta a aquesta pregunta de diferents formes:

### 2.1 Origen

Tot i que el terme Intel·ligència Artificial no es va utilitzar per primer cop fins l'any 1956 per John McCarthy en la que va ser la primera conferència acadèmica sobre el tema, molts situen l'origen en un article de recerca publicat l'any 1950 pel matemàtic anglès Alan M. Turing, considerat per molts el pare de la informàtica moderna.

L'article, que portava per títol "*Computing Machinery and Intelligence*" [17], començava amb el plantejament de la pregunta "*Les màquines poden pensar?*" i en ell es proposava el famós Test de Turing, que es convertiria en un objectiu a llarg termini per la recerca en IA.

El test, o "*Imitation Game*" com s'anomena a l'article, es plantejava com un test simple capaç de demostrar que una màquina podia pensar. El seu plantejament pràctic consisteix en que un avaluador humà mantingui una conversació textual amb una màquina i una persona alhora, sense saber amb qui parla en cada moment. Si l'avaluador no és capaç de distingir si les respostes que rep són de la màquina o de la persona, llavors la màquina és conscient i pot pensar.

Tot i que el test ha servit com a objectiu a vèncer per molts investigadors durant els inicis del camp de la IA, es pot veure fàcilment que el fet de dissenyar una màquina capaç d'enganyar a un avaluador no és suficient per demostrar que té consciència i és intel·ligent, sinó que simplement és bona simulant que pot parlar. Aquesta idea és la que defensa John Searle en un article publicat l'any 1980 [5] on es descriu l'experiment mental anomenat *The Chinese Room Argument*, un dels arguments més coneguts de la filosofia moderna que rebutja el Test de Turing.

Per tant, des d'un punt de vista teòric i potser més filosòfic, podem definir la IA com una branca de la informàtica que intenta donar una resposta afirmativa a la pregunta plantejada per Turing i, per tant, replicar la intel·ligència en les màquines, tot i que no estigui del tot clar què vol dir que una màquina sigui intel·ligent.

## 2.2 Categories

### 2.2.1 IA forta

També anomenada Intel·ligència Artificial General (AGI), és una forma teòrica d'IA que té com a objectiu crear màquines intel·ligents que no es distingeixin de la ment humana. Però, de la mateixa forma en què aprèn que un nen, la màquina d'intel·ligència artificial hauria d'aprendre mitjançant aportacions i experiències, progressant i avançant constantment en les seves habilitats amb el pas del temps, sent per tant capaç d'aprendre i planificar el seu futur.

Avui en dia, però, no existeixen exemples pràctics de sistemes de IA forta, tot i que es poden citar persones com Marvin Minsky que han sigut altament optimistes en el que podríem aconseguir en unes quantes dècades en el camp de la IA. D'altres, en canvi, dirien que ni tan sols es poden desenvolupar sistemes d'IA forts. El problema principal per desenvolupar aquests sistemes recau, en part, en que les mesures d'èxit com la intel·ligència o la comprensió no estan explícitament definides.

### 2.2.2 IA dèbil

També anomenada Intel·ligència Artificial Estreta (ANI), és la forma pràctica de la IA avui en dia. Aquest tipus d'IA es centra en realitzar una tasca específica amb un alt rendiment i per aconseguir-ho, operen amb moltes més limitacions i restriccions que una ment humana bàsica, per aquest motiu, podríem dir que com a molt arriben a simular ser intel·ligents.

Alguns autors també parlen de l'existència de sistemes d'IA híbrida, que combinen tècniques de la IA dèbil amb l'accés a una quantitat massiva de dades al núvol. Ted Greenwald situa l'arribada d'aquest tipus de IA en el 2011 [8], amb la parella iPhone 4S/Siri com a primer exemple en la història.

Per tant, des d'un punt de vista pràctic, podem definir la IA com un conjunt de tècniques (algorismes) que, imitant el comportament humà, ens permeten donar sol·lucions a problemes molt específics i potencialment complexos.

## 2.3 Conceptes clau

Per tal d'entendre l'àmbit pràctic del projecte és important diferenciar els conceptes IA, Machine Learning i Deep Learning, ja que s'acostumen a utilitzar indistintament per a referir-se a programes intel·ligents i no són el mateix. A mode de resum, podem veure a través de la Figura 2.1 que el *deep learning* és un subconjunt de *machine learning* i aquest alhora de la IA.

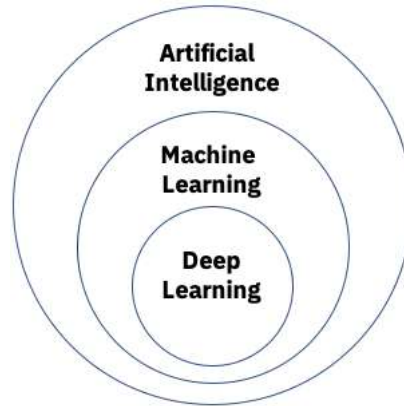


Figura 2.1: IA vs ML vs DL

### 2.3.1 IA

Com ja hem explicat, podem entendre la IA com qualsevol programa capaç de simular el comportament humà i donar sol·lucions a problemes d'una forma que considerem "intel·ligent".

Si observem la Figura 2.1, el fet que el ML sigui un subconjunt de la IA vol dir per tant que existeixen sistemes de IA que no utilitzen *machine learning*, com és el cas dels anomenats sistemes experts, que intenten simular la presa de decisions que faria un expert d'un camp concret, com per exemple els assistents virtuals d'incidències. Aquest és un exemple de IA del tipus GOF AI (*Good Old-Fashioned Artificial Intelligence*), més comú en els començaments del camp de la IA però encara present avui en dia en empreses com Arago [15], caracteritzat per tenir el coneixement directament inserit pels desenvolupadors, comunament codificat com un seguit de sentències condicionals.

### 2.3.2 Machine Learning

El *machine learning* és una tècnica present en la majoria de sistemes de IA moderns degut al seu alt potencial en termes de rendiment i la seva facilitat d'implementació, ja que el coneixement ara no ha de ser inserit directament sinó que, mitjançant un conjunt de dades prou gran i una metodologia estadística, el sistema és capaç d'aprendre i extreure coneixement per tal de prendre decisions informades a partir d'aquestes dades.

Els algorismes que utilitzen aquesta tècnica es poden diferenciar en dos tipus segons el tipus d'aprenentatge que duen a terme:

- **Aprenentatge supervisat:** les mostres del conjunt de dades estan etiquetades, fent possible que es puguin classificar en diferents classes o estudiar les relacions entre les etiquetes presents. S'anomena supervisat ja que gràcies a les etiquetes es pot comprovar que la classificació o les relacions que fa són correctes, i ajustar els paràmetres si el resultat és poc precís.
- **Aprenentatge no supervisat:** el conjunt de dades no està etiquetat, l'algorisme és capaç de trobar relacions ocultes entre les dades sense la intervenció humana, per aquest motiu s'anomena no supervisat. S'utilitza en problemes d'agrupació de dades, d'associació o de reducció de dimensió.

### 2.3.3 Deep Learning

El terme *deep learning* fa referència a la tècnica d'aprenentatge que utilitza xarxes neuronals amb

una quantitat alta de capes ocultes. És per aquest motiu reben el nom d'aprenentatge profund, perquè els valors d'entrada segueixen un camí molt llarg a través de la xarxa fins a generar un valor de sortida. Aquests conceptes es presenten en la secció de Xarxes Neuronals 2.4, més endavant.

A part, mentre qualsevol algorisme de *machine learning* necessita analitzar les dades segons la forma en que l'hem programat per fer-ho, és a dir, el desenvolupador ha de dur a terme el procés de *feature extraction* de forma manual, els algorismes de *deep learning* són capaços d'adaptar-se sols i descobrir les *features* en les dades sense haver-los programat específicament per trobar-les, podent aprendre per sí mateixos davant de patrons diferents de dades.

Per *feature* entenem una propietat individual i mesurable, característica d'un fenomen, que ens permet discriminar les diferents classes en el conjunt de dades. Per veure-ho amb un exemple, si disposem d'un conjunt de mostres que recull dades mèdiques de diferents pacients i volem poder predir si una persona desenvoluparà una malaltia cardiovascular, probablement l'índex de massa corporal o la presència d'accidents cardiovasculars en la família podrien arribar a ser dades que tindran un pes important en la predicció, i per tant, possibles *features* a tenir en compte. En el cas d'imatges és més complicat ficar un exemple, ja que una *feature* probablement serà un valor concret d'un grup de píxels en alguna zona concreta de la imatge.

## 2.4 Xarxes neuronals

L'estructura i el nom de les xarxes neuronals estan inspirats en el cervell humà, simulant la forma que tenen les neurones biològiques d'enviar-se senyals unes a altres.

L'origen de les xarxes neuronals resideix en un article publicat per Warren S. McCulloch i Walter Pitts l'any 1943 [12], d'on va néixer la idea que les neurones es podien codificar com a variables binàries de l'àlgebra de Boole que representarien l'activació o desactivació de cada neurona. Tot i que aquesta idea es va anar perfeccionant durant les properes dècades, és important saber que aquestes tècniques ja s'utilitzaven en projectes de recerca als anys 80 [9], però no es van fer realment populars fins que els processadors es van fer més potents, al voltant dels anys 2000, amb el naixement del *deep learning*.

### 2.4.1 Neurones

Una xarxa neuronal està formada per nodes, o neurones artificials, que es troben agrupades en capes i tenen la següent estructura:

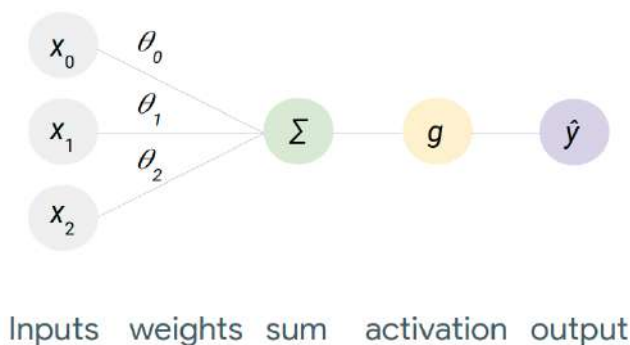


Figura 2.2: Exemple d'una neurona

Com podem veure en la Figura 2.2, les diferents  $x_i$  representen valors d'entrada a la neurona, cadascun amb un pes  $\theta_i$  associat, per tal de codificar la importància de cada entrada en particular. Aquests valors llavors es sumen i es comparen amb un llindar constant d'activació  $g$ , si aquest valor no es supera, la neurona no s'activa i no envia el seu valor de sortida a la resta de neurones de la xarxa a les quals està connectada. L'anterior representació es pot entendre matemàticament com:

$$\hat{y} = \sum_{i=0}^3 x_i \theta_i + g \equiv (x_0 \theta_0 + x_1 \theta_1 + x_2 \theta_2) + g$$

És important adonar-se que gràcies al valor d'activació aconseguim que la funció no sigui lineal i, d'aquesta manera, s'aconsegueix que la combinació de moltes neurones sigui efectiva [2]. D'altra forma, si es combinessin moltes neurones que calculen la seva sortida de forma lineal, es podria sintetitzar tota la xarxa en una sola neurona ja que, la suma de funcions lineals dona com a resultat una funció lineal i el que busquem es poder modelar un coneixement més complex que una regressió lineal bàsica. En altres paraules, la majoria dels problemes del món real pels quals s'utilitzen les xarxes neuronals no són linears, sinó que es necessari introduir la no-linearitat per tal de representar-los.

### 2.4.2 Xarxa neuronal profunda

Com ja s'ha comentat, les neurones d'una xarxa neuronal s'agrupen en capes, com podem veure en la Figura 2.3. Mentre que totes tenen una capa d'entrada i una de sortida, el nombre de capes ocultes que continguin pot variar i es considera que una xarxa neuronal amb dos capes ocultes o més ja passa a ser una xarxa neuronal profunda [6].

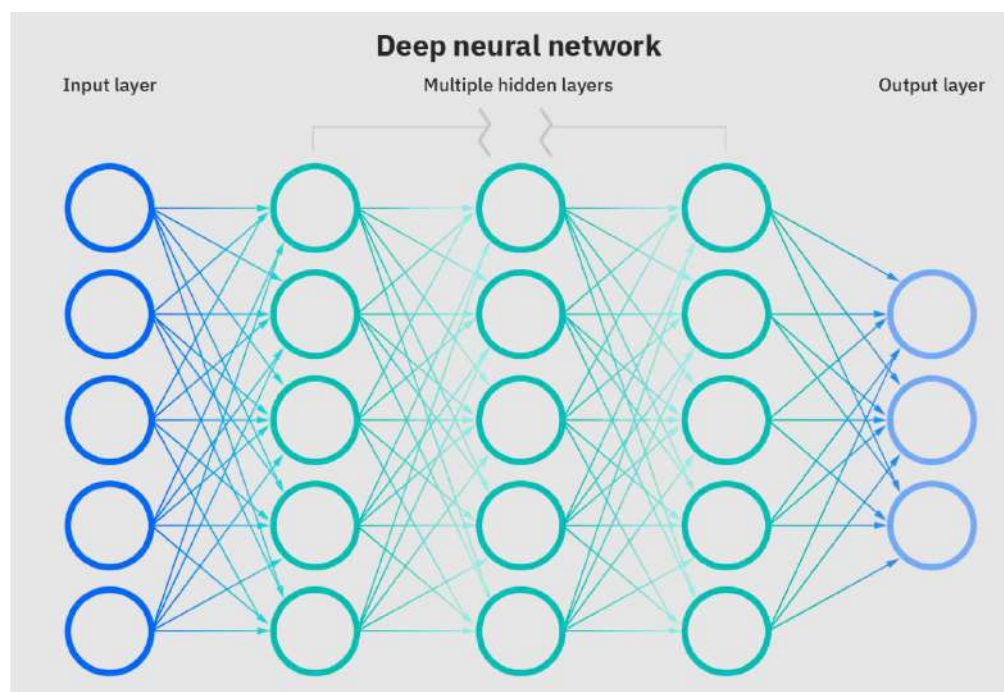


Figura 2.3: Estructura d'una Xarxa Neuronal

A més, la xarxa representada en la Figura 2.3 és exemple d'una xarxa neuronal **densament connectada** ja que en totes les capes cada neurona està connectada amb totes les neurones de la capa següent.

### 2.4.3 Xarxa neuronal convolucional

Aquest tipus de xarxa neuronal és la base de la recerca en visió artificial d'avui en dia (*computer vision*) i s'anomena així perquè aplica la tècnica de convolució, un filtre de reducció de la dimensió de la imatge que permet simplificar la tasca d'extreure'n *features*, és a dir, determinar en quins punt de la imatge s'hi troben els aspectes més rellevants.

Per tal d'explicar com funciona el filtre, primer s'ha de veure que una imatge es pot representar mitjançant una matriu on la mida correspondria a la resolució de la imatge i els valors de cada cel·la serien la codificació del color en aquell píxel en RGB. Per tant, d'aquesta forma podem representar imatges en color amb 3 matrius de la mateixa mida, una per cada canal de color RGB.

El filtre s'estableix com una matriu (*kernel*) que es centrarà sobre tots els píxels possibles de la imatge original, generant un valor pel nou píxel filtrat que serà el producte escalar entre el *kernel* i aquella secció de la imatge. Veient la Figura 2.4 es pot entendre més clarament la relació entre els píxels d'entrada i els de sortida:

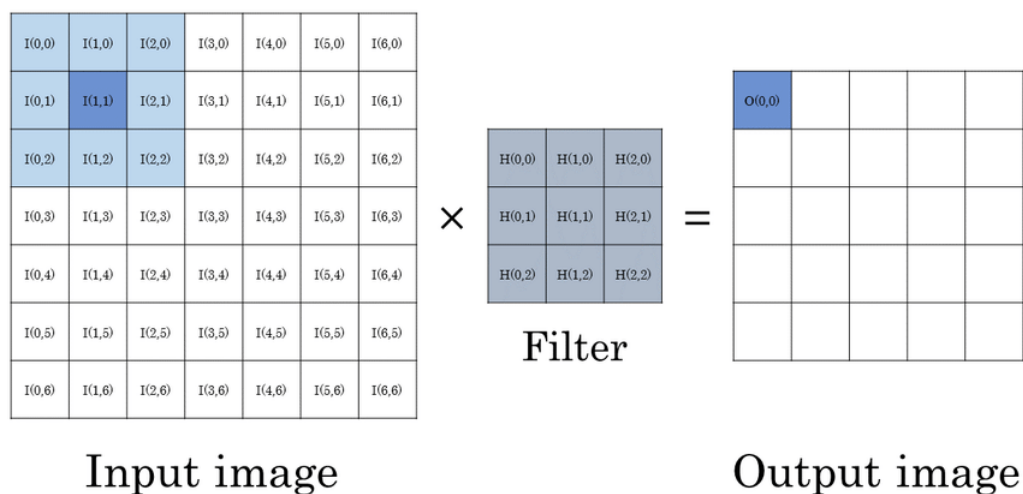


Figura 2.4: Filtre de convolució

El càlcul que duu a terme aquest filtre és el següent, on  $f$  és la imatge original de mida  $(m, n)$  i  $h$  és el *kernel* de mida  $(j, k)$ :

$$Conv(f, h) = \sum_j \sum_k h_{jk} \cdot f_{(m-j)(n-k)}$$

El que permet aquest filtre és que cada píxel de la imatge filtrada mantingui informació sobre els píxels que la imatge original tenia al voltant, però amb una mida reduïda, permetent simplificar la cerca de *features*.

En quant a la mida del *kernel*, aquesta sol ser senar ja que els *kernels* amb mida parella fan que es perdi massa informació de la imatge inicial, perquè els píxels centrals no tenen representació en la imatge reduïda. A més, en temes de visió artificial, la informació menys rellevant sol trobar-se als marges de les imatges mentre que preservar la informació central sol ser més important, de forma general. Es pot veure en la Figura 2.5 com es perd aquesta informació quan es treballa amb un *kernel* de mida parella:

131	162	232
104	93	139
243	26	252

131	162
?	?
104	93

Figura 2.5: Pèrdua d'informació d'un kernel de mida 2x2, a l'esquerra la imatge original i a la dreta la imatge filtrada

## 2.5 Fases d'un projecte de *Machine Learning*

El desenvolupament de qualsevol projecte de *machine learning* es pot estructurar en les següents quatre fases principals:

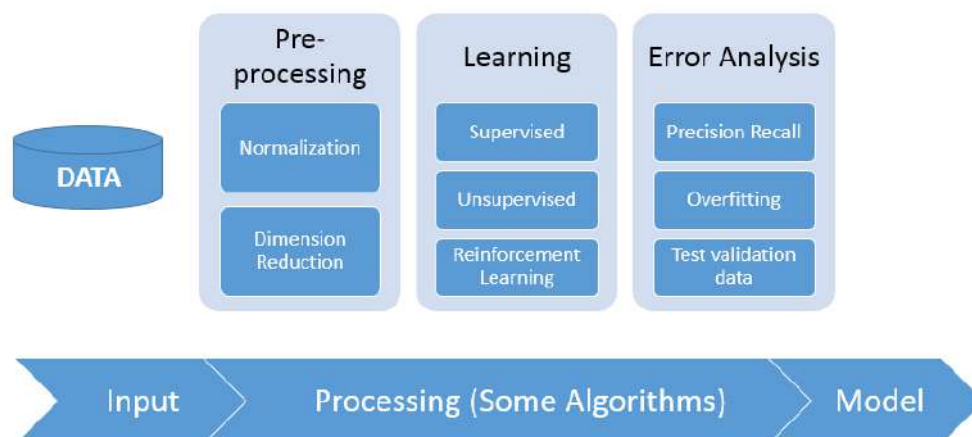


Figura 2.6: Fases bàsiques d'un projecte de ML

- **Obtenció de dades:** és el pas més important del procés del qual en dependrà la viabilitat i la major part de l'èxit, cap projecte de *machine learning* es pot dur a terme sense una quantitat considerable de dades. També és un pas difícil de realitzar per un individu sol si es desitja treballar amb dades reals, ja que es requereix d'una infraestructura prèvia de recollida de dades. Davant d'aquest inconvenient, la millor opció és treballar amb conjunts de dades de lliure accés.

A part de la quantitat, també és important la qualitat de les dades, ja que la presència d'inconsistència en les dades, la falta de coherència i els valors que falten, entre d'altres, generen problemes que després són difícils de detectar, tot i que existeixen tècniques de tractament de dades per tal de preparar-les i corregir-los si no es poden evitar.

- **Preprocessament i anàlisi de les dades:** abans de començar a treballar amb les dades és essencial analitzar-les per tal d'entendre la seva organització i representació. D'aquesta forma, es pot detectar l'existència d'errors i es poden aplicar tècniques per corregir-los.

Depenent del tipus de dades, tant si són numèriques com categòriques, és molt probable que sigui necessari normalitzar-les, és a dir, modificar la representació dels seus valors per tal que es pugui treballar amb valors d'una escala adequada.

Un cop les dades ja estan tractades, cal separar-les en dos conjunts com a mínim: un d'entrenament i un de test. El primer s'utilitzarà per tal que el model aprengui a predir les característiques que ens interessa observar en les dades i el segon servirà per comprovar si el model resultant té un bon funcionament, avaluant-lo sobre mostres a les quals mai ha tingut accés abans.

- **Construcció i entrenament del model:** un cop s'ha entès quin és el tipus de problema que es vol resoldre s'ha de triar el model adequat per fer-ho, construint-lo des de zero o utilitzant models que s'hagin utilitzat per resoldre problemes similars i hagin estat entrenats prèviament. L'avantatge d'utilitzar models prèviament entrenats és que es pot estalviar temps i recursos reaprofitant els bons resultats d'un model que ja funciona.

Quan ja es disposa del model, s'ha d'entrenar amb les dades que disposem. Entrenar un model simplement vol dir exposar-lo a un conjunt de dades per tal que pugui determinar uns valors adequats per aquells paràmetres modificables dels quals disposa, que permetin obtenir la menor pèrdua mitjana entre tots els exemples que ha consultat, entenent per pèrdua la distància que separa la predicció incorrecta del model al valor real que hauria d'haver predit. En altres termes, determinar els valors que permetin maximitzar la precisió del model.

- **Avaluació del model:** quan el model ha après per sí mateix els valors que permeten maximitzar la precisió de les prediccions, s'ha de comprovar que el model funciona com s'espera. Amb aquest objectiu, s'exposa el model a un conjunt de dades que no hagi vist prèviament, a fi de comprovar que el model no s'ha entrenat de forma errònia i només és capaç de predir les dades que ja coneix, que seria el que es coneix com a **overfitting**, un dels problemes més comuns en el desenvolupament de models de *machine learning*. Es refereix a que el model ha après a predir les dades d'entrenament massa bé, tant que ha après al detall, fent que sigui incapaç de realitzar prediccions correctes sobre mostres que s'allunyin lleugerament d'aquests detalls.

Un cop analitzem el rendiment sobre el conjunt de test del model, hem de decidir si la precisió amb la qual fa inferència (genera prediccions sobre dades noves) es pot millorar fent modificacions al model o si hem d'aplicar estratègies diferents per evitar l'*overfitting* en cas de tenir-ne. En aquest moment és quan es necessari determinar en quin punt del desenvolupament s'ha de tornar per tal d'aplicar canvis i generar un nou model.

## 2.6 Aplicacions en *Computer Vision*

Com que està clar que el model d'aquest projecte haurà de treballar amb un conjunt de dades d'imatges, a continuació presentarem algunes aplicacions del *deep learning* en el camp de la visió artificial:

- **Classificació d'imatges:** consisteix en assignar una etiqueta a la totalitat d'una imatge segons en el contingut que s'hi mostri. Un exemple que s'utilitza comunament com a referència de



prova seria la classificació del conjunt de dades MNIST, un conjunt de dades popular que conté imatges de nombres de l'1 al 10 escrits a mà. Dins d'aquest tipus de problemes tenim:

- Classificació binària: només hi ha dos etiquetes a assignar, per exemple decidir si en una radiografia hi ha un tumor o no.
- Classificació multi-classe: les etiquetes a assignar són tres o més. Per exemple diferenciar de quin tipus de vehicle és la foto: cotxe, moto, autobús, camió, etc.
- Detecció d'objectes: aquesta tipologia de problema és una versió més difícil que l'anterior ja que ara cal detectar l'objecte i a més indicar on es troba dins la imatge, envoltant-lo amb una rectangle que marqui els límits on està contingut.
- Segmentació d'objectes: aquest problema es basa en trobar les línies presents en una imatge, per tal de segmentar-ne les parts segons el seu significat semàntic. Són exemples de segmentació d'objectes: diferenciar els diferents plans en una fotografia, trobar el límit entre els arbres i el cel en la fotografia d'un paisatge, diferenciar els límits de cada persona en una imatge on s'abracen dues persones, etc.
- Seguiment d'objectes: consisteix en definir la localització d'objectes en la imatge, com en la detecció d'objectes, excepte que en aquest cas s'ha de mantenir el seguiment dels objectes detectats a través d'una seqüència d'imatges. Aquest seria el tipus de seguiment fan algunes càmeres de vigilància d'aeroports, capaces de fer un reconeixement facial i seguir el desplaçament de les persones, per exemple.
- Generació d'imatges: aquestes tècniques tenen un impacte més important en aspectes artístics majoritàriament i consisteixen en crear una imatge nova a partir de les imatges d'entrada. Els següents tipus es poden entendre com filtres ja que no fan avaluació d'objectes:
  - Transferència d'estil: consisteix en aprendre l'estil d'un artista o l'estil d'un conjunt de dades i transferir-lo a una imatge nova. El model recollit en l'article "*A Neural Algorithm of Artistic Style*" [7] aplica estils d'artistes com Pablo Picasso o Vincent van Gogh a imatges de paisatges.
  - Colorització d'imatges: consisteix en convertir una imatge en escala de grisos a una imatge en color. L'article "*Colorful Image Reconstruction*" [19] presenta un algoritme capaç de coloritzar imatges, amb exemples de personatges com Elvis Presley.
  - Reconstrucció d'imatges: és la tasca d'emplenar parts d'una imatge que falten o estan malmeses. Se'n pot veure un exemple en l'article "*Image Inpainting for Irregular Holes Using Partial Convolutions*" [11].

## 3. Disseny del projecte

En aquest apartat s'explicarà quines decisions s'han pres per tal de donar resposta als objectius referents al disseny del projecte plantejats en l'apartat 1.2, així com quins objectius no es poden satisfer en aquesta fase del desenvolupament del projecte i quedaran, per tant, pendents de ser satisfets en fases posteriors.

### 3.1 Anàlisi de Requeriments

- Per tal de classificar i definir el problema, caldrà investigar i estudiar quines patologies són les més viables de ser diagnosticades a través d'un model d'IA. En l'apartat 3.2 s'aclareix la viabilitat de diagnosticar el foc bacterià i es presenta també la opció finalment escollida.
- Amb l'objectiu de trobar un conjunt de dades per entrenar el model caldrà investigar quines opcions hi ha disponibles de forma oberta en la xarxa, així com quines són les fonts oficials que hi ha disponibles en el territori. En l'apartat 3.2 s'explica quin conjunt de dades s'ha escollit finalment i en el capítol 4 se'n fa una anàlisi en més detall.
- Per tal de desenvolupar el model, caldrà triar l'entorn d'entrenament adequat, amb la capacitat de computació suficient per poder dur a terme les tasques més pesades en termes de càlcul, com són l'entrenament i la validació. Es pot trobar una anàlisi dels diferents entorns d'entrenament que s'han sospesat per dur a terme el projecte a l'apartat 3.3.
- Finalment, per poder justificar la validesa del model serà necessari realitzar experiments amb diferents tipus de models, realitzant modificacions en el paràmetres configurables disponibles i comprovant-ne el rendiment sobre el mateix conjunt de prova, aquest punt quedarà pendent de ser satisfet en l'apartat de modelat 6.

### 3.2 Obtenció de dades

Inicialment, la idea del projecte era el desenvolupament d'un model de classificació d'imatges capaç de diagnosticar la presència de foc bacterià en les imatges, però aquest plantejament s'ha trobat amb dos problemes principals:

- **El diagnòstic en laboratori és imprescindible per diagnosticar el foc bacterià [13].** Després d'investigar quins símptomes caracteritzen la malaltia, s'ha comprovat que tots aquells símptomes visuals són fàcilment confusibles amb d'altres malalties i que, per tant, no era possible basar el diagnòstic només en l'aspecte visual. A més, totes les tècniques de diagnòstic existents impliquen l'anàlisi de mostres a un nivell molecular, serològic o d'espectrometria de rajos [3], dades que no es poden recollir en imatges.

- **Falta de fonts fiables amb un conjunt de dades suficientment gran.** Tot i la impossibilitat de realitzar un diagnòstic només basat en imatges, en aquest punt de la recerca es considerava l'opció d'utilitzar una combinació de dades meteorològiques amb imatges per tal d'avaluar els riscos d'aparició de la malaltia, seguint models de predicció que ja haguessin estat provats i validats en el terreny [18].

Després de descartar les fonts *online* davant la falta de fiabilitat i la impossibilitat de verificar el diagnòstic en les mostres, s'ha realitzat una consulta a les oficines del Departament d'Agricultura, Ramaderia, Pesca, Alimentació i Medi Natural (DARP) a Lleida. Malauradament, la quantitat i la qualitat de les dades obtingudes tampoc han possibilitat la implementació d'un model d'aquestes característiques.

Per aquests motius, davant les problemàtiques presentades, s'ha optat per descartar el foc bacterià i donar prioritat a la qualitat de les dades a l'hora de fer la cerca de la nova malaltia a diagnosticar.

Finalment s'ha escollit el conjunt de dades de la competició *Plant Pathology 2021 - FGCV8* [16], la vuitena edició d'un taller de problemes de visió artificial de gra fi, organitzat per diferents universitats i entitats tecnològiques com Google, Caltech i Cornell.

El conjunt de dades conté imatges de fulles de poma amb diferents malalties que es mostren al capítol 4 en més detall. En quant al disseny, cal remarcar que el projecte intentarà resoldre un problema de **classificació d'imatges de gra fi**, ja que no volem classificar objectes amb una etiqueta genèrica, com seria diferenciar si un objecte és una fulla o un arbre, sinó que volem classificar tipus concrets de fulles a partir de diferències menys pronunciades que en la classificació genèrica d'objectes.

### 3.3 Entorns d'entrenament

Un dels grans reptes d'aquest projecte ha sigut trobar un entorn adequat per realitzar l'entrenament, ja que personalment mai havia treballat amb un conjunt de dades tant gran i amb unes tècniques que requerissin tanta capacitat de computació. A continuació es resumeixen els diversos entorns d'entrenament que s'han valorat durant el desenvolupament del projecte, així com els motius pels quals s'han acabat descartant.

#### 3.3.1 Local

##### 3.3.1.1 Combinació de CPU i GPU

Inicialment, l'enfoc era entrenar el model tal i com s'havia fet en altres treballs d'IA duts a terme al llarg del grau, de forma local i amb els propis recursos de l'ordinador personal. A més, després de buscar informació i descobrir que TensorFlow tenia l'opció d'executar-se fent ús de la GPU, semblava lògic pensar que aquesta opció seria viable. La targeta gràfica amb la qual s'han realitzat les proves és una Nvidia GeForce GTX 960M, amb una capacitat de computació CUDA de 5.0. Com a referència, es pot dir que va sortir al mercat al gener de 2015 i actualment la que té una capacitat de computació més alta és la GeForce RTX 3090, de 8.6 i posada a la venda al setembre de 2020.

Tot i així, un cop configurat l'entorn, en el moment d'entrenar el model aquesta opció tardava 20 hores per model i mai va arribar a acabar l'entrenament de cap model sencer ja que, després de la primera hora, la memòria de la targeta gràfica (2GB) s'emplenava i l'execució es cancel·lava.

### 3.3.1.2 Nvidia Jetson Nano 2GB TPU

Després del fracàs d'intentar utilitzar els recursos propis de l'ordinador i com que, afortunadament, disposava d'una unitat de processament tensorial o TPU, vaig realitzar proves per veure si era viable dur a terme l'entrenament en aquesta unitat especialitzada en el processament de càlculs per a TensorFlow.

Després d'instalar la distribució específica d'ubuntu en una targeta microSD connectada a la unitat i poder utilitzar les mostres de models d'exemple que vénen amb el joc de desenvolupador, vaig adonar-me que la distribució no tenia instal·lats cap dels paquets que se solen utilitzar en el desenvolupament de projectes d'IA, ni tan sols Python. Per aquest motiu, vaig indagar més en la forma en què els desenvolupadors utilitzaven aquest producte i vaig descobrir que tot i que era possible usar-lo com tenia planejat inicialment, el producte estava dissenyat específicament per fer inferència de forma eficient i no per a servir com a entorn d'entrenament, és a dir, està ideat principalment per a robòtica en sistemes que necessitin fer inferència de forma molt ràpida, com per exemple podrien ser sistemes autònoms.

Això afegit al fet que la memòria microSD de la qual disposava no era prou gran com per contenir el conjunt de dades, el sistema operatiu i els diversos paquets de desenvolupament a la vegada, i també tenint en compte la complexitat d'adaptar aquest entorn amb una distribució específica, va provocar que aquesta opció s'acabés descartant i s'encarés la cerca d'un entorn al núvol.

## 3.3.2 Cloud

Inicialment s'havia descartat la opció de dur a terme l'entrenament al núvol ja que aquests serveis, tot i que són més potents, no solen ser gratuïts i de fet tenen un cost elevat per a usuaris individuals, ja que estan dissenyats per a empreses. A continuació es presenten les diferents opcions gratuïtes que s'han valorat.

### 3.3.2.1 Google colab

Google Colaboratory o colab és un servei de notebook basat en el projecte Jupyter que no requereix configuració per utilitzar-lo i brinda accés gratuït a recursos computacionals, incloses GPU (Tesla K80). Tot i que està especialment enfocat a projectes d'IA, presenta també diversos inconvenients:

- No disposa d'emmagatzemament persistent, els arxius s'esborren quan es reinicia la sessió, fent que sigui realment incòmode treballar amb conjunts de dades grans, com és el cas d'aquest projecte. Per solucionar-ho s'ha intentat fer servir Google Drive però només ofereix 15GB d'emmagatzemament gratuït i el conjunt de dades n'ocupa 18GB.
- S'apaga després d'una hora sense interacció per part de l'usuari, parant el notebook i fent inviable dur a terme entrenaments més llargs.
- A diferència de Kaggle, que veurem més endavant, no disposa d'un mecanisme de control de versions.

Sembla per tant una bona opció per projectes de mida més reduïda però davant la impossibilitat de tenir el conjunt de dades en la seva totalitat, s'ha descartat aquesta opció.

### 3.3.2.2 Google cloud

Google Cloud és una plataforma que aglutina un conjunt d'aplicacions de desenvolupament web, des d'emmagatzemament, gestió d'APIs, computació, anàlisi de dades, entre d'altres. Està fortament enfocat a empreses i no és un servei gratuït, tot i que ofereixen 300 dòlars de crèdit gratuït a gastar durant el transcurs de tres mesos, amb un conjunt de serveis reduït i limitant-ne la funcionalitat.

Després d'iniciar la prova gratuïta s'ha comprovat que la complexitat d'aquesta eina és més alta del que inicialment es pensava, ja que amb tants serveis en oferta és difícil trobar l'adequat pel projecte, perquè la majoria de serveis d'IA ofereixen eines de prova on l'algoritme d'entrenament està predefinit o no es poden pujar més de 500 imatges a la vegada. A part, el conjunt de dades tampoc hi cap al límit gratuït de 5GB mensuals de Google Cloud Storage (GCS).

Per aquests motius i també tenint en compte que la TPU més barata costa 5 dòlars l'hora, és a dir, que es tindrien només 60 hores útils de TPU dins la prova gratuïta sense comptar el que costaria emmagatzemar el conjunt de dades, s'ha decidit descartar aquesta opció amb preocupació també pels possibles sobre costos afegits un cop esgotat el crèdit gratuït.

### 3.3.2.3 Kaggle

Kaggle és una comunitat online on es celebren competicions d'IA, s'ofereixen cursos, conjunts de dades gratuïts i on els enginyers de dades poden discutir i compartir codi entre ells. Va ser fundada el 2011 i adquirida per Google el 2017. El gran avantatge sobre la resta d'opcions és que Kaggle ja disposa del conjunt de dades que es vol utilitzar en aquest projecte, per la qual cosa no cal pujar-lo. De fet, està allotjat en un bucket de Google Cloud Storage accessible a través de la pròpia llibreria de datasets de Kaggle.

Similar a com funciona Google Colab, Kaggle ofereix kernels basats en Jupyter i amb accés a 30h gratuïtes d'acceleradors TPU i GPU que es renoven cada 5 dies. A part, també compta amb un sistema integrat de control de versions que permet guardar l'estat del kernel amb el que s'està treballant, per tal que altres usuaris puguin veure una execució concreta.

Finalment s'ha triat aquest entorn per la simplicitat que ofereix així com pel fet que l'accés al conjunt de dades és el més convenient d'entre totes les opcions disponibles. En la següent taula 3.1 es pot veure un resum de les característiques de cada entorn d'entrenament.

Taula 3.1: Comparativa d'entorns d'entrenament

Entorns	Característiques			
	Hardware	Emmagatzemament	Preu	Velocitat/model
CPU + GPU local	GeForce GTX 960M	1 TB HDD	≈200€	20h
TPU local	128-core Maxwell	32 GB MicroSD	≈65€	-
Google Colab	Cluster TPUv2	15 GB Drive	Gratuït	-
Google Cloud	Cluster TPUv2	5 GB GCS	5\$/hora	-
Kaggle	Cluster TPUv2	Autogestionat, GCS	Gratuït	1h 30min

## 4. Anàlisi exploratori de les dades

El conjunt de dades conté 18.632 imatges de fulles de poma, amb una resolució de 4000x2672 píxels. Conté mostres de fulles sanes, amb òxid, amb crostes, floridures, la taca foliar (*Cercospora sojina*) i mostres complexes que designen diagnòstics amb múltiples malalties presents. Les classes en anglès són: healthy, rust, scab, powdery mildew, frog eye leaf spot i complex, respectivament.

### 4.1 Distribució de les classes

A continuació podem veure com es troben distribuïdes les 12 classes diferents al llarg del dataset:

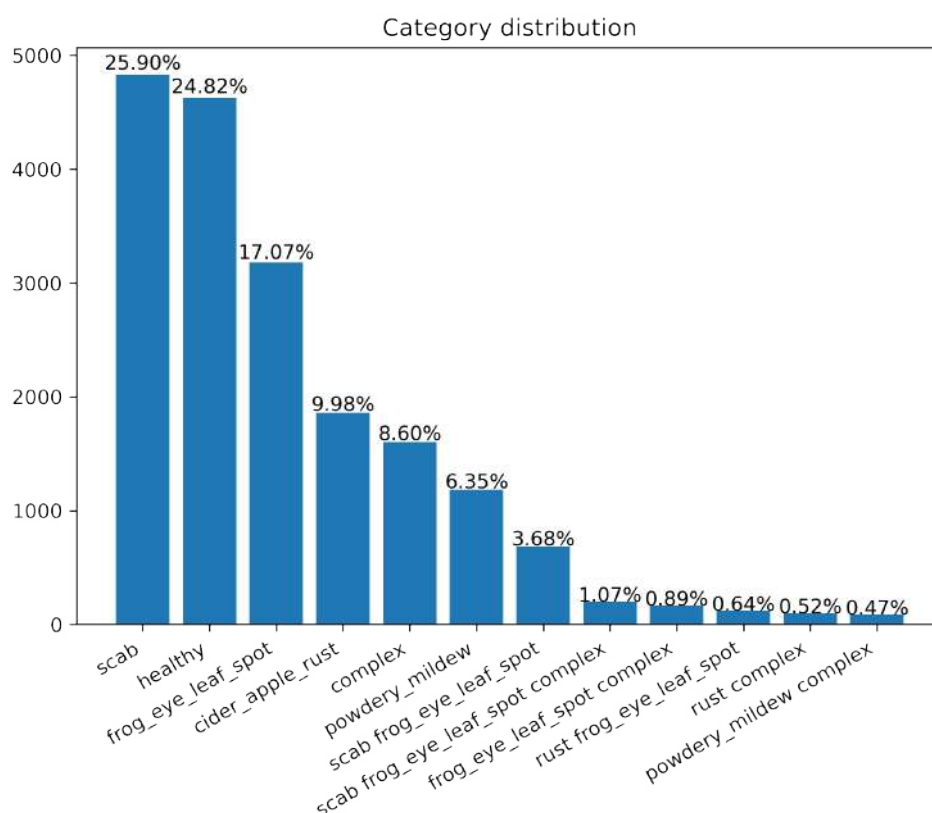


Figura 4.1: Distribució inicial

Com podem observar en la Figura 4.1, les imatges que presenten combinacions de les malalties bàsiques estan inicialment representades com a classes diferents i això provoca que existeixi un gran desequilibri de mostres en el dataset, fet que podria provocar els següents problemes: per una banda podria ser que el model no pogués predir correctament aquestes classes amb poca

representació, per la falta de mostres a partir de les quals extreure coneixement. És el problema que es coneix com a undersampling. Per l'altra banda, podria ser que intentés extreure tot el coneixement possible a partir d'un conjunt tan reduït de mostres que no assolís l'objectiu principal d'extreure un coneixement genèric de totes, fent que el model només fos precís en classificar les poques mostres que coneix. En aquest cas es tindria un problema d'overfitting i el model seria igualment incorrecte.

De fet, això és justament el que ha passat quan s'ha entrenat el model amb aquesta mateixa distribució, en aquest cas amb la xarxa neuronal DenseNet121. Com podem veure en la figura 4.2, ben aviat el model ha arribat a precisions elevades, del 90% o més, arribant fins al 99% de precisió sobre el conjunt d'entrenament i el 90% sobre el de validació. Valors tan elevats solen ser indicadors de que el model no funciona correctament i, de fet, realitzant proves s'ha observat que era altament inconsistent ja que, tot i classificar la majoria de mostres correctament, aquelles que classificava incorrectament ho feia amb un 99% de certesa.



Figura 4.2: Precisió amb la distribució inicial

Per evitar aquest tipus de problemes és important que inicialment les diferents etiquetes del dataset que es voldran predir estiguin equilibrades i per aquest motiu s'ha decidit representar la informació del conjunt de dades de forma diferent, aquest plantejament s'explica de forma estesa en la documentació del notebook de Kaggle del projecte. Ara, enlloc que cada imatge només tingui una etiqueta associada, en cas que una imatge tingui per etiqueta combinacions de classes bàsiques, se li associarà etiquetes diferents per cada classe bàsica, fent així que el dataset es pugui representar només amb les classes principals i quedant de la següent forma visible en la Figura 4.3.

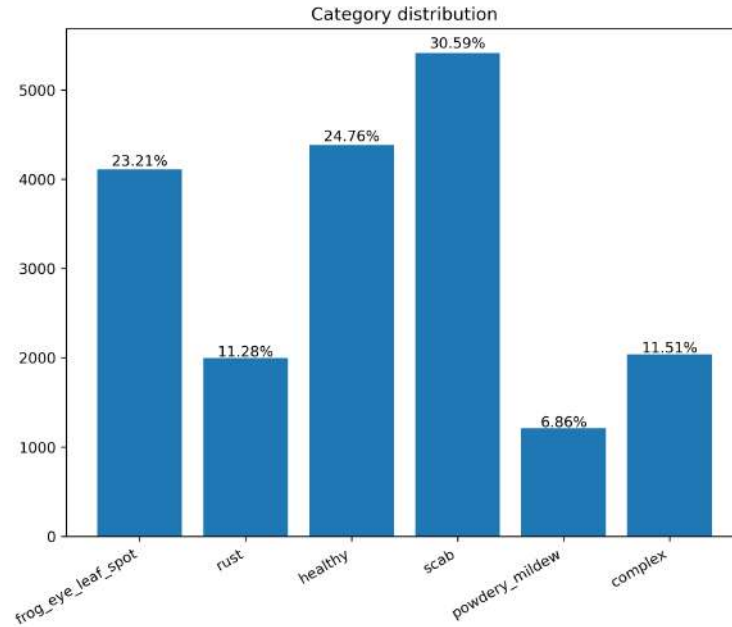


Figura 4.3: Distribució corregida

D'aquesta forma, ara es pot observar que les classes no estan tant desequilibrades en quant a la representació, evitant possibles problemes durant l'entrenament.

## 4.2 Visualització de mostres

A continuació es mostren en la Figura 4.4 alguns exemples de les diferents imatges presents en cada classe. A partir d'aquesta mostra podem observar que no totes les imatges tenen la mateixa resolució i que, per tant, caldrà normalitzar les imatges a una mida estàndard. També es pot observar que hi ha imatges que estan mal etiquetades, com és el cas de la tercera imatge amb scab, on es pot veure clarament que hauria d'estar etiquetada com a healthy.

Aquest errors, anomenats outliers, són difícils de detectar perquè tot i que existeixen mecanismes per trobar-los, la tasca es especialment difícil de realitzar en imatges ja que comporta l'entrenament d'un model a part dedicat a la detecció d'outliers [4]. Degut a la complexitat de detectar-los no s'ha aplicat aquesta tècnica en la realització d'aquest projecte.

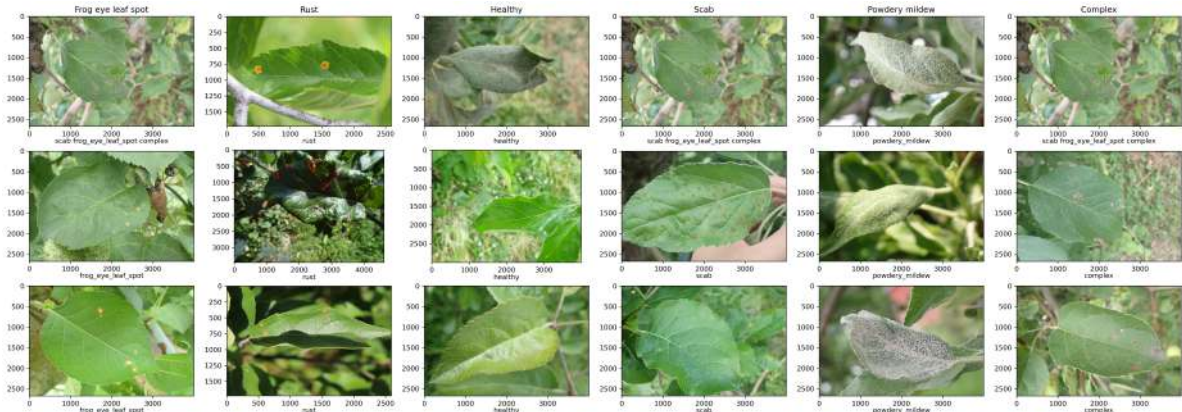


Figura 4.4: Mostres per classe



## 5. Preprocessament

En el preprocessament es duen a terme aquelles tasques que preparen el conjunt de dades per tal que l'algoritme pugui treballar correctament amb elles. Normalment entre aquestes tasques s'inclouen la normalització del dataset, l'eliminació de dades incorrectes o la generació de dades, entre d'altres. A continuació s'expliquen les diferents tècniques de preprocessament que s'han utilitzat:

### 5.1 Normalització

Normalitzar les dades significa expressar-les sobre una escala regulada de valors, de forma que les relacions entre elles es mantingui. Aquest sol ser el cas quan s'han de normalitzar valors categòrics com podria ser per exemple el conjunt de valors [sí, no] en [1, 0]. La normalització que s'ha dut a terme en el cas d'aquest projecte ha sigut la separació de les diferents etiquetes, que en el conjunt inicial es trobaven com a valors d'una única columna anomenada "labels", a columnes independents per cada etiqueta on els valors 1 i 0 representen si aquella malaltia és present en la imatge o no, respectivament. Cal remarcar, però, que altres normalitzacions eren també possibles, com representar cada etiqueta amb un valor diferent i utilitzar una única columna.

### 5.2 Eliminació de duplicats

S'ha utilitzat la llibreria `imagehash` per tal de trobar imatges duplicades amb l'objectiu d'eliminar-les, ja que no aporten informació nova i fan que l'entrenament es repeteixi de forma arbitrària per a certes classes, afectant un cop més a la distribució balancejada d'aquestes, ja que causen *overfitting* per subconjunts petits del model.

S'ha utilitzat un algoritme de *Perceptive Hashing* per trobar aquests duplicats, que es pot entendre com una funció de hash que permet comparar imatges, ja que se s'utilitzen les features presents en la imatge per generar una empremta diferent per cada imatge, però no única per a imatges similars, permetent que les empremtes siguin comparables. D'aquesta forma s'aconsegueix avaluar les similituds entre les imatges i, per fer-ho, aquests algorismes treballen analitzant les freqüències baixes de la imatge, ja que és on en resideix l'estructura. Les característiques principals d'aquest tipus d'algorismes a tenir en compte és que són ràpids de calcular i poden trobar similituds entre imatges iguals però de mida diferent, amb relacions d'aspectes diferents o amb canvis menors en el color, com variacions de contrast o brillantor.

El resultat obtinguts després d'haver aplicat aquesta tècnica de preprocessament es troben a l'apartat "Eliminació de duplicats" 6.3.2 d'aquest projecte, on es corrobora l'augment de la precisió del model després d'utilitzar-la.

### 5.3 Ajustar la resolució de les imatges

Com s'ha comentat abans, les diferents imatges del dataset tenen resolucions diferents, a diferència de com es pensava inicialment. Per aquest motiu és important treballar amb una mida fixa, per tal que el model pugui tractar-les de la mateixa forma. Aquesta també ha sigut una tasca de normalització però aquest cop sobre les imatges i no sobre les etiquetes. S'ha decidit treballar amb imatges de 512x512 píxels, tot i que és un paràmetre més que es podria variar per tal de comprovar l'impacte en el rendiment final.

### 5.4 Preparació dels conjunts de test, entrenament i validació

Com s'ha explicat en l'apartat 2.5, cal separar el conjunt de dades inicial en un conjunt d'entrenament i un conjunt de test com a mínim, per tal de poder avaluar el model sobre mostres a partir de les quals no s'ha entrenat. En el nostre projecte destinarem un 10% d'imatges al conjunt de test, el 90% de mostres restants s'usaran per entrenar els models. Aquesta partició s'ha fet de forma pseudoaleatòria per tal de mantenir una distribució similar entre les classes i amb una llavor concreta, per tal que l'experiment sigui reproducible. A més, s'utilitza un conjunt de validació format per un 15% de les mostres del conjunt d'entrenament, que utilitzarà l'algoritme d'entrenament a cada iteració de l'entrenament (epoch), per tal d'avaluar com evoluciona el rendiment del model entrenat fins llavors, ja que intentarà trobar valors òptims per incrementar la precisió i reduir la pèrdua, mesurats sobre el conjunt de validació.

### 5.5 Data Augmentation

La generació i augmentació de dades és una tècnica molt útil que aporta grans avantatges en qualsevol projecte d'intel·ligència artificial, els principals sobretot són que permeten corregir dades que inicialment estaven incompletes i obtenir dades noves a partir de les dades inicials, que permeten augmentar la mida del conjunt d'entrenament i regularitzar el dataset per tal d'evitar l'overfitting i, per tant, augmentar la precisió del model. És una pràctica relativament nova en el camp de la visió artificial i actualment està àmpliament acceptada ja que s'ha demostrat que pot arribar a incrementar la precisió de forma considerable inclús en projectes que disposen de moltes dades. De fet, un estudi realitzat el 2017 a l'Universitat de Stanford demostra que la precisió en un model que utilitzava data augmentation augmentava fins a un 6% respecte al mateix model que no utilitzava aquesta tècnica [14].

Cal però remarcar que no és recomanable utilitzar aquesta tècnica amb l'objectiu de balancejar la distribució entre les diferents classes, ja que si per exemple, es decideix generar imatges desenfocades a partir de les imatges originals només en aquelles classes on la representació es baixa, el model no estarà generalitzant ja que com que només té mostres d'imatges desenfocades per aquella classe, sempre que trobi una imatge desenfocada la classificarà directament com aquella classe. Per aquest tipus de problema és preferible retallar les classes amb més representació o generar duplicats de les classes amb menys representació.

S'han utilitzat dos transformacions per tal d'augmentar el dataset: rotar la imatge de forma aleatòria en l'eix vertical i rotar-la de forma aleatòria en l'eix horitzontal. Amb aquestes transformacions simples s'aconsegueix generar dades noves a partir de les quals el model pot extreure coneixement.

## 6. Modelat

Per tal de mostrar quins models existents s'han utilitzat en aquest projecte, primer és important explicar les tècniques bàsiques que constitueixen els models més populars en la comunitat: convolució en dues dimensions, MaxPooling i activació per ReLU. És important dir també que la majoria de models importants en el camp de la visió artificial s'han entrenat amb imatges de ImageNet, un projecte que té com a objectiu proveir projectes de recerca amb grans datasets d'imatges i compta amb més de 14 mil·lions d'imatges de 20.000 classes diferents.

Com que el funcionament de la convolució en dues dimensions ja s'ha explicat en l'apartat 2.4, a continuació es procedeix a explicar en què consisteix el MaxPooling i l'activació per ReLU.

El codi relacionat amb l'entrenament és pot trobar en el notebook de Kaggle<sup>1</sup> associat al projecte i el codi utilitzat per la generació de resultats, proves i gràfiques, al repositori de Github<sup>2</sup>.

### 6.1 MaxPooling

La tècnica de MaxPooling és molt similar a la convolució i també s'utilitza per reduir la complexitat de les xarxes neuronals convolucionals mentre es retenen els features presents en la imatge. Aquest cop, però, enlloc de triar un kernel i fer el producte escalar, la tècnica de MaxPooling consisteix en trobar el valor màxim dins d'una finestra de la matriu original. Cal dir, a més, que també es comú utilitzar l'AveragePooling que, enlloc de quedar-se amb el valor màxim dins d'una finestra, fa la mitja entre tots els valors d'aquella finestra. O el GlobalPooling, que retorna el valor màxim en tota la matriu, com seria el 112 en el cas de la Figura 6.1. I per últim, el GlobalAveragePooling, que retornaria la mitja de tots els valor en la matriu sencera. Aquestes són diverses tècniques de pooling que s'utilitzen de forma usual en xarxes neuronals profundes.

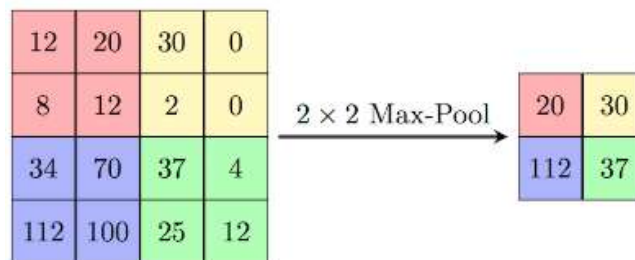


Figura 6.1: Maxpooling amb una finestra de 2x2

<sup>1</sup>Kaggle notebook: [kaggle.com/guillemfelis/plant-pathology-2021](https://kaggle.com/guillemfelis/plant-pathology-2021)

<sup>2</sup>GitHub Repository: [github.com/gfelis/TFG](https://github.com/gfelis/TFG)

El càlcul que duu a terme el MaxPooling es pot expressar com la següent funció, on  $f$  és la imatge original de mida  $(m, n)$  i  $h$  és la finestra de mida  $(j, k)$ :

$$MaxPool(f, h)_{mn} = \max(m + j - 1, n + k - 1 \ \forall \ 1 \leq m \leq l, 1 \leq n \leq l)$$

Per tal d'extreure els features, el que fan aquestes xarxes neuronals és combinar capes de convolució amb capes de MaxPooling, amb diferents mides de finestra o kernel, per tal d'anar reduint la dimensió de la imatge original fins que al final l'aplanen en una sola dimensió i la processen.

## 6.2 ReLU

ReLU (Rectified Linear Unit) és una funció d'activació comunament utilitzada en arquitectures de xarxes neuronals. Fa referència al valor d'activació  $g$  de les neurones, explicat en l'apartat 2.4 d'aquest document, que determina en quines circumstàncies el valor d'una neurona s'activa i s'envia a les següents capes de la xarxa neuronal. ReLU és una funció no lineal molt simple que permet augmentar la capacitat de modelatge dels models convolucionals. Visualment, es pot veure el seu funcionament en la Figura 6.2. El que fa és no activar les neurones que donin valors de sortida negatius i activar només aquelles que en donin de positius.

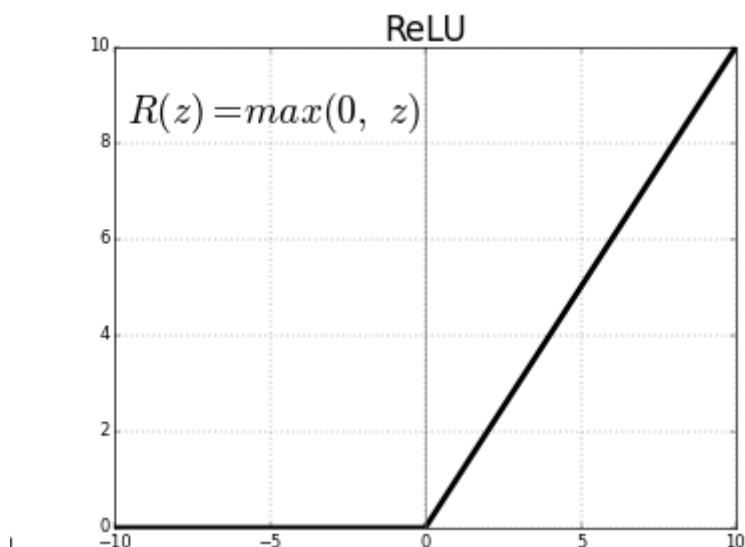


Figura 6.2: Activació per ReLU

Un cop més podem expressar aquesta funció de forma formal com:

$$ReLU(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$$

## 6.3 DenseNet

### 6.3.1 Arquitectura del model

El primer model que s'ha utilitzat ha sigut DenseNet121. Mentre que les xarxes convolucionals tradicionals amb  $L$  capes tenen  $L$  connexions, una entre cada capa i la seva capa posterior,

les xarxes DenseNet tenen  $L(L + 1) / 2$  connexions directes. Per a cada capa, els mapes de característiques (feature maps) de totes les capes anteriors s'utilitzen com a entrades i els seus propis mapes de característiques s'utilitzen com a entrades a totes les capes posteriors. Les xarxes DenseNet tenen diversos avantatges convinents: alleugeren el problema del gradient de desaparició, reforcen la propagació dels features, fomenten la reutilització de les features i redueixen substancialment el nombre de paràmetres. L'article de recerca que presentava el 2018 aquest tipus de xarxes [10] va evaluar la seva arquitectura proposada en quatre tasques de referència (benchmarks) de reconeixement d'objectes altament competitives (CIFAR-10, CIFAR-100, SVHN i ImageNet). Les xarxes DenseNets van obtenir millores significatives respecte a les altres xarxes competents en la majoria d'ells, mentre que requereixen menys memòria i càlcul per aconseguir un alt rendiment. A continuació es mostra la seva arquitectura de forma esquemàtica:

Com podem observar en la Figura 6.3, la xarxa està principalment composta per capes de convolució en dues dimensions, activació per ReLU i MaxPooling en dues dimensions també. A més, hi trobem altres tipus de capes com la de ZeroPadding que el que afegeix columnes i files de zeros als marges de la imatge per tal d'intentar conservar la informació a les vores, fent que quan s'aplica convolució i maxpooling els valors no es perdin. A part, també trobem capes de BatchNormalization que s'utilitzen per ajustar la mida de la entrada per a la següent capa, per a cada batch, que es pot traduir com a lot.

Per explicar què és un batch s'ha de dir que aquests algorismes d'aprenentatge funcionen de forma iterativa, actualitzant diverses vegades els paràmetres entrenables dels que disposen a cada iteració. A cadascuna d'aquestes iteracions se l'anomena epoch i cada epoch s'estructura a la vegada en batches, que són el subconjunt d'imatges del conjunt d'entrenament que processa el model en cada epoch abans d'actualitzar els paràmetres. Ficant un exemple, si disposem d'un conjunt d'entrenament de 10.000 mostres i fem batches de 200 imatges, tindrem un total de 5.000 batches. Si entrenem el model en 20 epochs, per cada epoch, després d'haver vist 200 imatges s'actualitzaran els paràmetres del model, arribant a actualitzar-se fins a 5.000 vegades per epoch o 100.000 vegades en total durant l'entrenament.

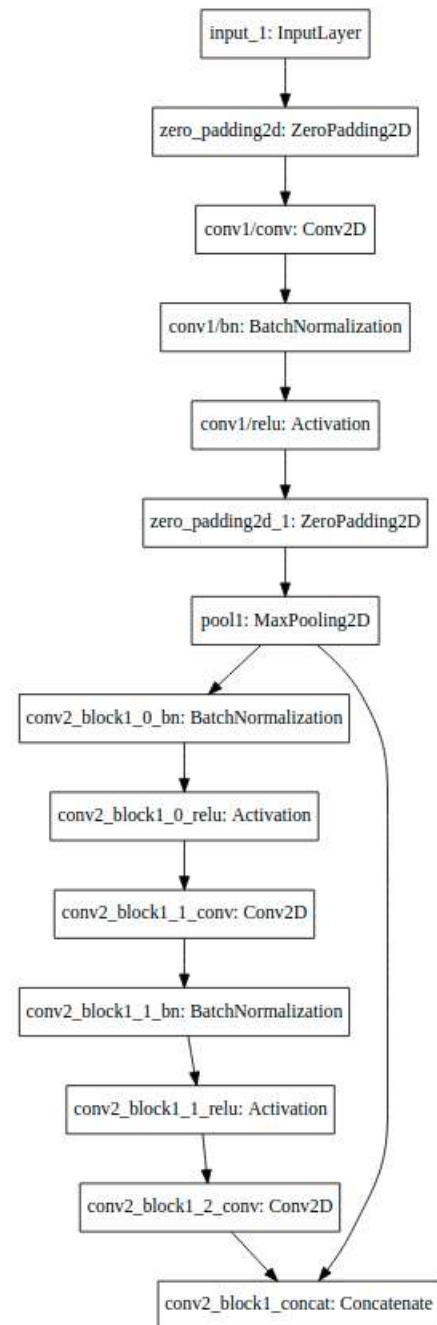


Figura 6.3: Blocs bàsics DenseNet121

En la Figura 6.4 tenim representada l'estructura del model final que s'ha utilitzat. La primera capa fa referència al model DenseNet121 vist anteriorment, al qual se li ha afegit una capa de global average pooling per tal de convertir la sortida de dimensió 16x16x1024 de l'anterior capa a un únic vector de 1024 valors. Finalment amb la capa densa final s'obté una sortida de 6 valors, un per classe, on cada valor representa la probabilitat que té una imatge d'entrada de ser classificada amb cada etiqueta.

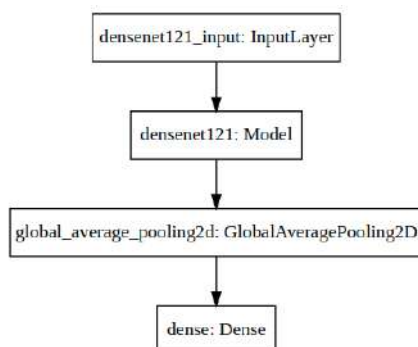


Figura 6.4: Arquitectura del model final

### 6.3.2 Resultats

Durant l'entrenament d'aquest model s'han obtingut les variacions en la precisió sobre el conjunt d'entrenament i de validació que es poden observar en la Figura 6.5. Cal remarcar que tot i un començament irregular, la precisió sobre el conjunt de validació ha obtingut bons resultats. La precisió d'aquest model s'estima en un 85% segons les dades recollides durant l'entrenament. Cal dir també que s'han reduït el nombre d'epochs de 20 a 15 respecte al primer entrenament (dut a terme amb les etiquetes separades) degut a l'estancament que presentava aquell model cap al final de l'entrenament i amb l'objectiu d'evitar-lo aquest cop.

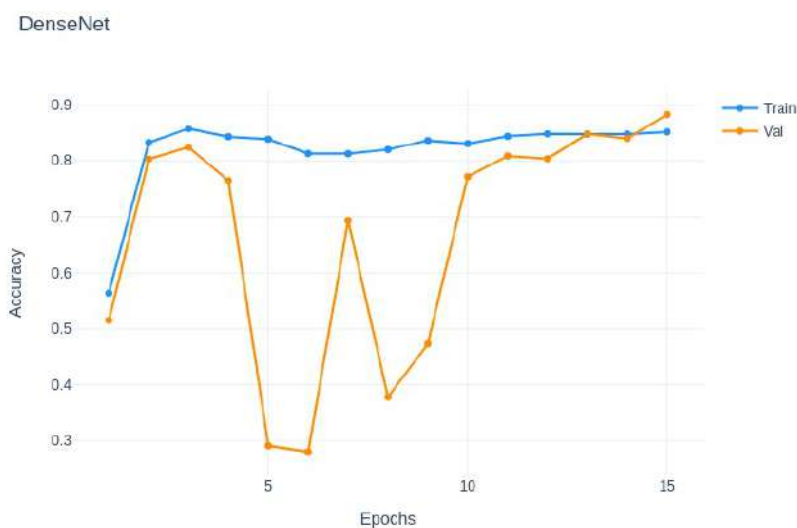


Figura 6.5: Evolució en la precisió sobre els conjunts d'entrenament i validació

En aquest moment cal decidir a partir de quin llindar es predirà que una imatge presenta una única etiqueta i quan serà la combinació de vàries. Inicialment, s'ha decidit que quan la probabilitat de la predicció més elevada fos inferior al 55%, s'optés per pensar que el model feia una predicció poc segura perquè es tractava d'un cas en que hi havia una segona etiqueta a tenir en compte i per tant, la predicció real era la combinació de la etiqueta amb la probabilitat més alta amb la segona etiqueta amb probabilitat més elevada. Els resultats obtinguts d'aquesta forma són els següents:

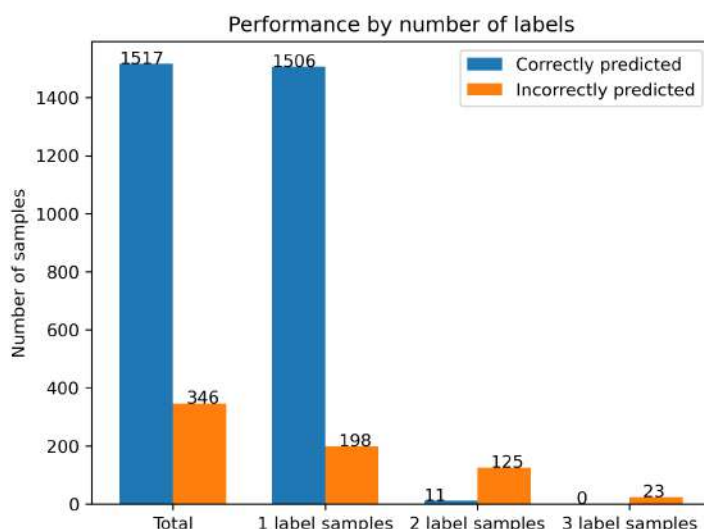


Figura 6.6: Classificació inicial, llindar del 55%

Cal dir també que realitzar aquest tipus de test sobre un conjunt del 10% de mida (1.836 imatges) ha tardat 25 minuts i 56 segons a completar-se, processant una imatge per segon aproximadament. Com es pot comprovar, la precisió general del model és del 82.62%, on en el cas de predir imatges amb una sola malaltia augmenta fins a un 88.38%, mentre que si es tracta de predir imatges amb dues malalties baixa fins al 8.09% i com que no hem previst el cas que una imatge pugui tenir 3 malalties, en aquest cas la precisió és del 0%.

Es pot, per tant, concloure que aquesta forma de decidir quan s'han de predir múltiples malalties no és vàlida. Per tal de trobar una forma millor s'ha tornat a realitzar una prova sobre el conjunt de test però aquest cop enlloc d'establir un llindar absolut s'establirà que predigui dues classes si la diferència entre la predicció més alta i la segona predicció més alta és inferior a un cert llindar, és a dir, si tenen probabilitats properes de ser les dues. També s'establirà un llindar per dur a terme una predicció de tres classes. D'aquesta forma, ara es basa l'elecció en la seguretat relativa a la resta de possibles prediccions.

Realitzant proves amb porcions més petites del conjunt de test, s'ha observat que les diferències entre les precisions en els casos en què la predicció correcta està formada per dues classes va des de del 5% fins al 40% i s'ha establert un llindar de prova inicial del 25%. En quant a la diferència entre la segona predicció més alta i la tercera, no s'ha pogut observar degut a la poca representació de mostres però s'ha detectat que la finestra és molt més petita i inclús una diferència del 10% no és suficient per millorar la predicció per aquests casos. S'ha de marcar també que augmentar aquesta finestra provoca que es perdin prediccions correctes per la classe on el model actualment té millor rendiment.

Tot i així els millor resultats s'han obtingut amb una finestra del 20% per a la primera diferència i d'un 10% per la segona, tot i que com es pot observar en la Figura 6.7 està clar que el model no fa la predicció de múltiples etiquetes de forma correcta, ja que només s'ha pujat la predicció del conjunt d'una etiqueta a un 89.2% i els de dues etiquetes a 9.56%, fent que el total sigui d'un 83.5%.

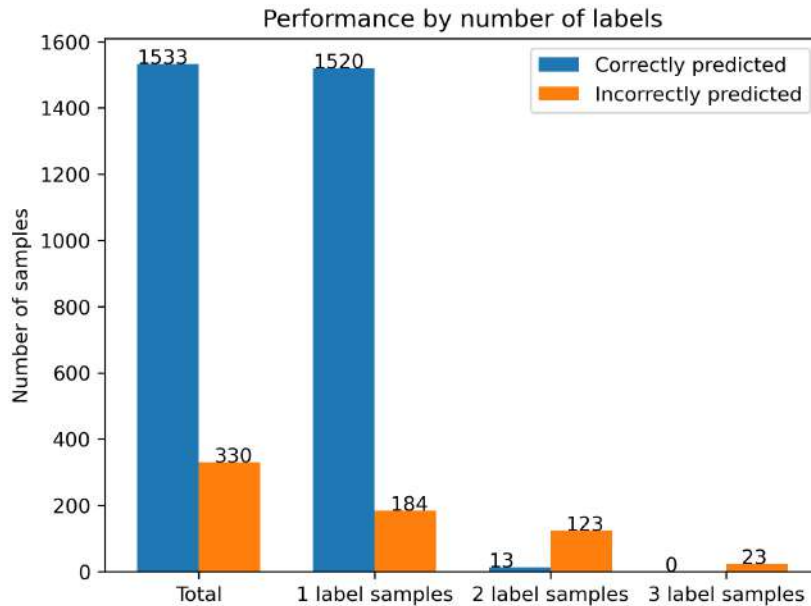


Figura 6.7: Distància relativa, llindar del 20% i 10% respectivament

Les alternatives que queden per millorar el rendiment seria trobar valors més òptims pels llindars, eliminar les mostres incorrectament etiquetades com s'explica en l'apartat 4 o provar d'afegir més tècniques d'augment de dades en l'entrenament i observar els resultats.

### Eliminació de duplicats

Amb la utilització d'aquesta tècnica de preprocessament s'han trobat 62 imatges repetides en el dataset, que sobre el total de 18.632 imatges representa l'eliminació d'un 0.33% de mostres del conjunt d'entrenament. El procediment seguit ha sigut l'eliminació de totes les parelles de mostres repetides en cas que les etiquetes associades no coincidissin, perquè eren casos de dades inconsistents i no es podia determinar quina era l'etiqueta correcta per aquella imatge. En cas que les etiquetes dels duplicats coincidissin, s'ha mantingut en el dataset una imatge d'aquella parella, per tal de no perdre informació. Un exemple dels duplicats que s'han trobat es pot observar en la Figura 6.8. D'aquesta forma, la primera imatge és eliminada completament mentre que de les altres dues se'n guarda una còpia en el conjunt d'entrenament.

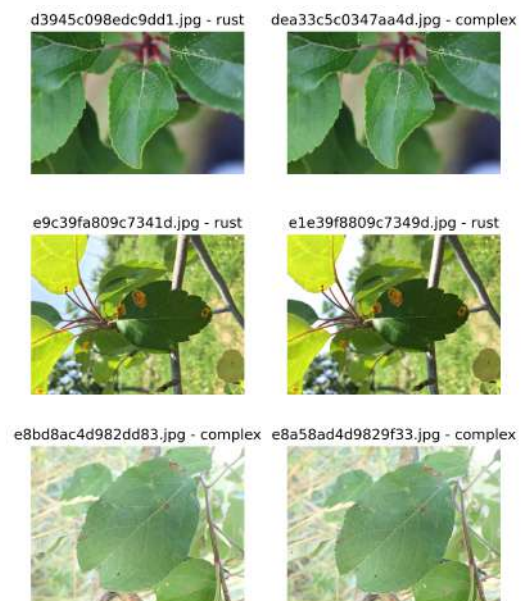


Figura 6.8: Mostra d'imatges duplicades



Amb la incorporació d'aquesta tècnica el model ha obtingut un 86.71% de precisió durant l'entrenament, amb un 90.11% sobre el conjunt de validació, representat a la Figura 6.9.

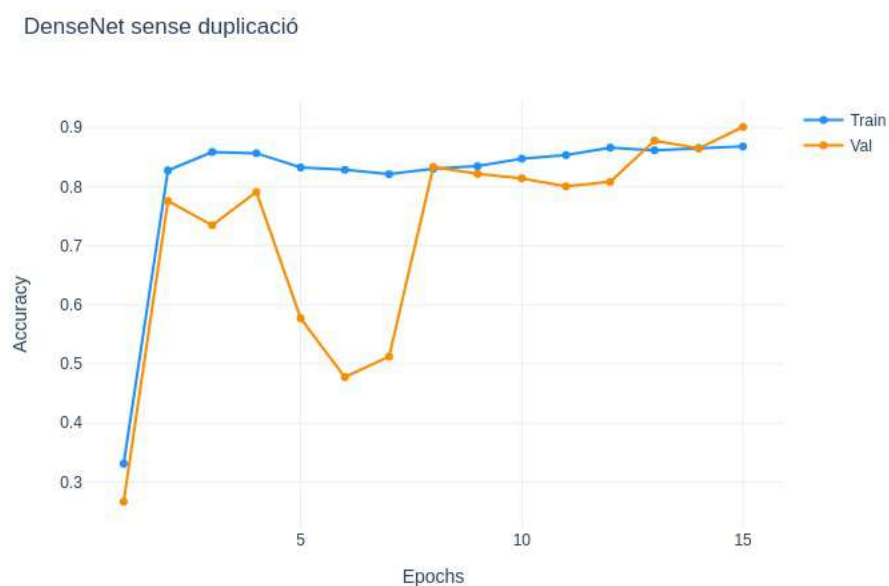


Figura 6.9: Rendiment durant l'entrenament de DenseNet, sense duplicació

Per tal de corroborar aquest augment en la precisió s'ha dut a terme de nou l'experimentació sobre el conjunt de test (Figura 6.10), on ha obtingut una precisió general del 87.36%, on la diferència més remarcable és que la precisió en el cas de 2 etiquetes ha passat del 9.56% al 35.3%, i inclús s'ha aconseguit un encert pel cas amb 3 etiquetes. Pel cas de predir mostres amb una etiqueta la precisió ha augmentat fins al 91.26%. Podem per tant concloure que és una tècnica molt útil per augmentar el rendiment tenint en compte sobretot la rapidesa amb què es pot implementar.

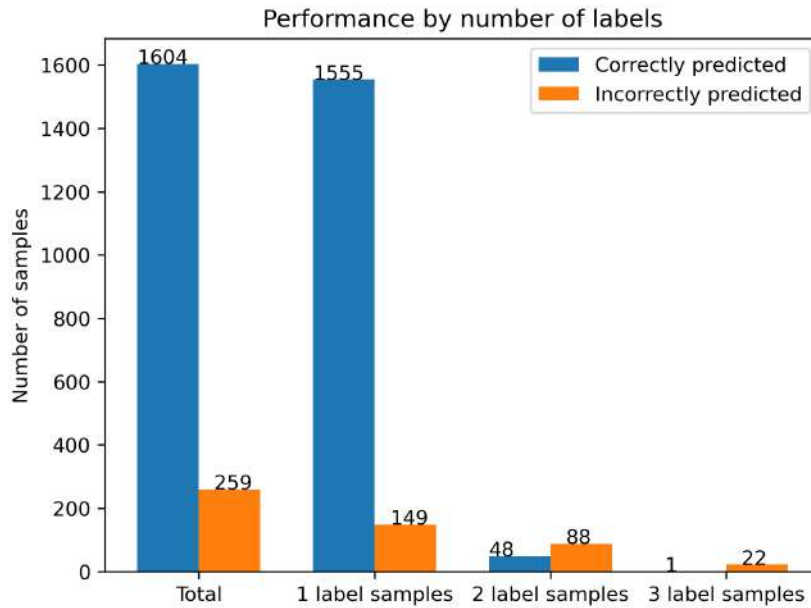


Figura 6.10: Rendiment sobre el conjunt de test

## 6.4 EfficientNet

Inicialment es planejava utilitzar dos models diferents d'EfficientNet: EfficientNet-B7 i EfficientNet-B7 Noisy Student, per tal de comparar resultats entre els diferent models i millorar les prediccions mitjançant la tècnica d'ensembling, que consisteix en generar una predicció final a partir de les prediccions de models diferents. Aquest enfoc, però, ha quedat greument afectat al comprovar la precisió que s'ha obtingut en entrenar el model amb els models pre-entrenats d'EfficientNet, que amb prou feines ha arribat a precisions del 55%, com es pot veure en la Figura 6.11 per al cas d'EfficientNet-B7, on l'entrenament és àmpliament irregular i s'estanca entorn al 50% des de bon principi, tot i havent seguit el mateix procés d'implementació que s'ha seguit per l'entrenament del model de DenseNet.

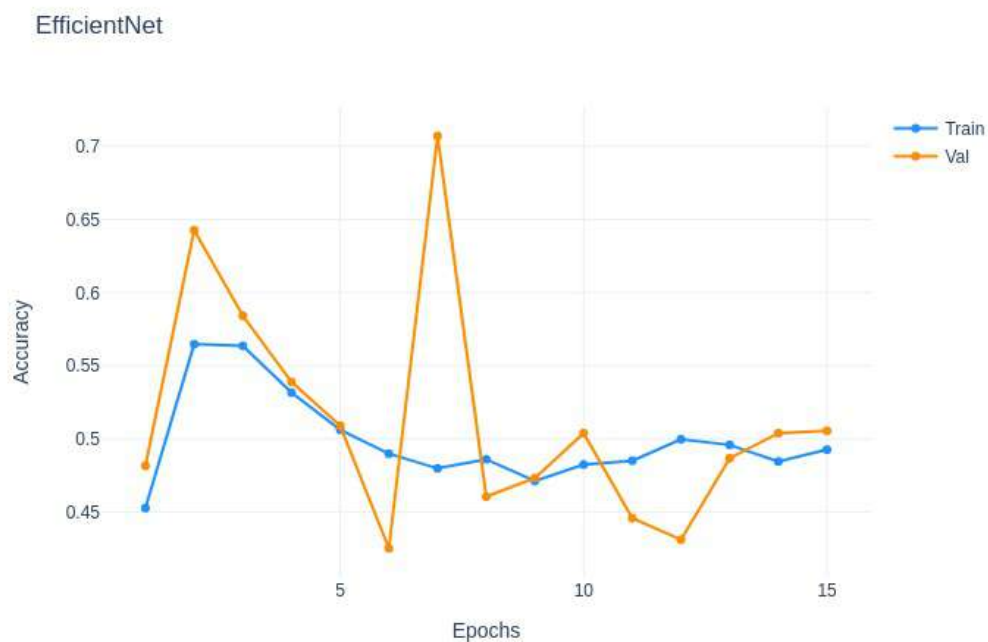


Figura 6.11: Evolució de la precisió sobre els conjunts d'entrenament i validació d'EfficientNetB7

És probable que aquest rendiment tant dolent sigui provocat perquè el model es troba estancat en un màxim local en el camí de trobar valors òptims per als paràmetres configurables de la xarxa des de bon començament.

A part d'això, EfficientNet és a més molt més pesada d'exportar i importar, 700MB contra els 80MB que pesa el model de DenseNet, fet que en dificulta també el treball ja que els recursos tant en local com en el notebook del cloud són limitats i pujar aquest model cada cop que es vol treballar amb ell ralentitza molt el desenvolupament.

Per aquests motius i veient els avantatges en rendiment que presenta el model de DenseNet, s'ha decidit abandonar el plantejament d'intentar combinar prediccions amb els resultats de l'entrenament amb aquests models.

## 7. Conclusions i futur del projecte

Des de bon inici, aquest projecte s'ha plantejat de forma molt ambiciosa, ja que contemplava la creació d'una aplicació d'Android amb el desplegament del model resultant de l'ensamblatge de 3 models diferents, entrenats per a detectar una malaltia de la qual no es tenien les dades ni la certesa que fos diagnosticable de forma visual. A més, la utilització d'unes tècniques tan avançades, noves i de les quals hi ha pocs professionals amb experiència degut a la novetat del camp, també ha causat que el procés d'aprenentatge dut a terme per tal d'implementar-les i justificar-les fos complexe alhora que motivador.

Per aquests motius, ha sigut molt necessari destinar temps a la recerca tant d'informació com de dades i metodologies per dur a terme la implementació, amb una tecnologia amb la qual no es tenia experiència i amb la qual també, segons la meva opinió, és necessària molta perícia per generar bons resultats. S'està parlant de tecnologies com són TensorFlow, Pandas, Keras, OpenCV i d'altres llibreries de l'àmbit de la ciència de dades. Per aquests motius també, s'ha fet evident la necessitat de reajustar els objectius inicials del projecte, que com en la majoria de projectes, han deixat de banda possibles característiques del resultat final per tal d'afavorir l'obtenció de resultats tangibles i efectius.

Entre les diverses concessions que s'han fet s'inclou la realització de més experimentació per tal d'intentar millorar el rendiment del model, ja que són tants els hiper-paràmetres configurables que s'ha optat per observar l'afectació en el rendiment variant només aquells que s'intuïa que tindrien un impacte major.

Tot i aquest ajustament dels objectius, cal mencionar que s'han obtingut uns resultats molt satisfactoris i s'ha pogut crear un model que utilitza DenseNet121 capaç de classificar correctament el 87.36% de mostres, a més, el procés seguit fins a arribar a aquest rendiment ha permès consolidar molts dels conceptes que en la fase de disseny del projecte quedaven difosos, com són l'extracció de features i la classificació de classes que poden interseccionar, i també s'ha pogut verificar que la utilització de tècniques com l'eliminació de duplicats augmenten altament el rendiment del model.

Taula 7.1: Evolució dels resultats

Model	Precisió				
	1 classe	2 classes	3 classes	Total	Observacions
EfficientNetB7 <sup>1</sup>	-	-	-	≈51%	Inviaible
EfficientNetB7-NoisyStudent <sup>1</sup>	-	-	-	≈54%	Inviaible
Denset121 v.0 <sup>1</sup>	-	-	-	99.99%	Overfitted
Denset121 v.1 <sup>2</sup>	88.38%	8.09%	0%	82.62%	-
Denset121 v.1.1 <sup>3</sup>	89.2%	9.56%	0%	83.5%	+1.12% total
Denset121 v.2 <sup>4</sup>	91.26%	35.3%	4.35%	87.36%	+3.86% total +25.74% 2 classes

<sup>1</sup> sense preprocessament de cap tipus

<sup>2</sup> amb preprocessament, expecte eliminació de duplicats, llindar 1 al 55%

<sup>3</sup> amb preprocessament, expecte eliminació de duplicats, llindar 1 al 25%, llindar 2 al 10%

<sup>4</sup> amb preprocessament, inclós eliminació de duplicats, llindar 1 als 25%, llindar 2 al 10%

En la figura 7.1 podem observar els diferents models utilitzats i com queda justificat l'ús de tècniques com la normalització del dataset, que ha permès balancejar la distribució entre les classes, així com la cerca d'uns llindars òptims a partir dels quals diferenciar el tipus de predicció i l'eliminació de duplicats que han permès augmentar el rendiment de forma significativa.

Tot i els bons resultats obtinguts, encara seria possible l'aplicació de tècniques de preprocessament i l'ajustament dels llindars de predicció per tal d'intentar augmentar la precisió, sobretot en el cas que s'hagi de fer una predicció amb intersecció de classes, ja que és la zona on hi ha més marge per millorar el rendiment. Quedaria, per tant, pendent de realitzar en el futur d'aquest projecte, així com la implementació del model en una aplicació d'android com inicialment s'havia considerat.

## 8. Bibliografia

- [1] Serge Savary et al. “The global burden of pathogens and pests on major food crops”. A: *Nature, Ecology and Evolution* 3 (2019), pàg. 430-439. DOI: <https://doi.org/10.1038/s41559-018-0793-y>.
- [2] Emma Amor. *Understanding Non-Linear Activation Functions in Neural Networks*. 2020. URL: <https://medium.com/ml-cheat-sheet/understanding-non-linear-activation-functions-in-neural-networks-152f5e101eeb>.
- [3] Nikrooz Bagheri et al. “Detection of Fire Blight disease in pear trees by hyperspectral data”. A: *European Journal of Remote Sensing* 51 (gen. de 2018), pàg. 1-10. DOI: 10.1080/22797254.2017.1391054.
- [4] Paul Bergmann et al. “MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection”. A: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pàg. 9584-9592. DOI: 10.1109/CVPR.2019.00982.
- [5] David Cole. “The Chinese Room Argument”. A: *The Stanford Encyclopedia of Philosophy*. Ed. d’Edward N. Zalta. Winter 2020. Metaphysics Research Lab, Stanford University, 2020.
- [6] IBM Cloud Education. *Neural Networks*. 2020. URL: <https://www.ibm.com/cloud/learn/neural-networks>.
- [7] Leon A. Gatys, Alexander S. Ecker i Matthias Bethge. *A Neural Algorithm of Artistic Style*. 2015. arXiv: 1508.06576 [cs.CV].
- [8] Ted Greenwald. *How Smart Machines Like iPhone 4S Are Quietly Changing Your Industry*. 2011. URL: <https://www.forbes.com/sites/tedgreenwald/2011/10/13/how-smart-machines-like-iphone-4s-are-quietly-changing-your-industry/?sh=1a4041e1598f>.
- [9] John Hopfield. “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”. A: *Proceedings of the National Academy of Sciences of the United States of America* 79 (maig de 1982), pàg. 2554-8. DOI: 10.1073/pnas.79.8.2554.
- [10] Gao Huang et al. *Densely Connected Convolutional Networks*. 2018. arXiv: 1608.06993 [cs.CV].
- [11] Guilin Liu et al. *Image Inpainting for Irregular Holes Using Partial Convolutions*. 2018. arXiv: 1804.07723 [cs.CV].
- [12] Warren S. McCulloch i Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. A: *Bulletin of Mathematical Biophysics* 5 (1943), pàg. 115-133.
- [13] Ana Palacio-Bielsa i Miguel A. Cambra Álvarez. *El fuego bacteriano de las rosáceas (Erwinia amylovora)*. 2a ed. NIPO: 770-09-348-2. Madrid: MARM, 2009.
- [14] Luis Perez i Jason Wang. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. 2017. arXiv: 1712.04621 [cs.CV].

- [15] Frank Säuberlich i Danko Nikolić. *AI without machine learning*. 2018. URL: <https://www.teradata.com/Blogs/AI-without-machine-learning#:~:text=In%20fact%2C%20today%20this%20type,an%20AI%20without%20machine%20learning..>
- [16] Ranjita Thapa et al. “The Plant Pathology Challenge 2020 data set to classify foliar disease of apples”. A: *Applications in Plant Sciences* 8.9 (2020), e11390. DOI: <https://doi.org/10.1002/aps3.11390>. eprint: <https://bsapubs.onlinelibrary.wiley.com/doi/pdf/10.1002/aps3.11390>. URL: <https://bsapubs.onlinelibrary.wiley.com/doi/abs/10.1002/aps3.11390>.
- [17] A. M. TURING. “I.—COMPUTING MACHINERY AND INTELLIGENCE”. A: *Mind* LIX.236 (oct. de 1950), pàg. 433-460. ISSN: 0026-4423. DOI: 10.1093/mind/LIX.236.433. eprint: <https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf>. URL: <https://doi.org/10.1093/mind/LIX.236.433>.
- [18] IPM University of California. *Fire Blight risk of apparition models*. 2020. URL: <http://ipm.ucanr.edu/DISEASE/DATABASE/fireblight.html>.
- [19] Richard Zhang, Phillip Isola i Alexei A. Efros. *Colorful Image Colorization*. 2016. arXiv: 1603.08511 [cs.CV].